

Content Documents Management

Ricardo Muniz

KENNEDY SPACE CENTER

Major: Computer Science

USRP Spring Session

Date: 9 MAY 11

Content Documents Management

Muniz R.1

University of Puerto Rico, Arecibo, Puerto Rico

Hochstadt J.2

University of Central Florida, Orlando, Florida

Boelke J.3 and Dalton A.4

Launch Control System, Kennedy Space Center, Florida

The Content Documents are created and managed under the System Software group with Launch Control System (LCS) project. The System Software product group is lead by NASA Engineering Control and Data Systems branch (NE-C3) at Kennedy Space Center. The team is working on creating Operating System Images (OSI) for different platforms (i.e. AIX, Linux, Solaris and Windows). Before the OSI can be created, the team must create a Content Document which provides the information of a workstation or server, with the list of all the software that is to be installed on it and also the set where the hardware belongs. This can be for example in the LDS, the ADS or the FR-1. The objective of this project is to create a User Interface Web application that can manage the information of the Content Documents, with all the correct validations and filters for administrator purposes. For this project we used one of the most excellent tools in agile development applications called Ruby on Rails. This tool helps pragmatic programmers develop Web applications with Rails framework and Ruby programming language. It is very amazing to see how a student can learn about OOP features with the Ruby language, manage the user interface with HTML and CSS, create associations and queries with gems, manage databases and run a server with MYSQL, run shell commands with command prompt and create Web frameworks with Rails. All of this in a real world project and in just fifteen weeks!

Nomenclature

LCS	= Launch Control System
NE-C	= NASA Engineering Control and Data Systems Division
OSI	= Operating System Image
Set	= operational or development room made up of hardware and software used to monitor and control end-items
LDS	= Launch Control System Development Set
ADS	= Application Development Set
FR-1	= Firing Room #2
Ruby	= a dynamic, reflective, general-purpose object-oriented programming language
Ruby on Rails	= often shortened by Rails or RoR, is a web application framework for the Ruby language.
OOP	= Object Oriented Programming
HTML	= Hyper Text Markup Language
CSS	= Cascading Style Sheets
Gems	= Packages that serves as library for the Ruby programming language
MYSQL	= a relational database management system that provides access to databases
NDC	= NASA Domain Control

1System Software Engineering Intern, NE-C3, Kennedy Space Center, FL.

2Information Architecture Intern, NE-C2, Kennedy Space Center, FL.

3Ricardo Muniz mentor and System Software Lead, NE-C3, Kennedy Space Center, FL.

4Jake Hochstadt mentor and Information Architecture Lead, NE-C2, Kennedy Space Center, FL.

Introduction

The LCS has as a goal to send commands from the different operational rooms to the shuttle. To achieve this there has to be various rooms known as sets and each set should have a numerous amount of workstations, not for only send commands, but to track the status of the shuttle. Each workstation has a number of software's which operate the different required tasks in order to have a successful launch. The System Software group is trying to get track of each software in every workstation of each set and finalize creating the OSI's, but to create this images there has to be a list of procedures to be done. One of these procedures is to create a list of Content Documents, which provides information of all the software's needed in a workstation. The problem of the NE-C3 branch is that they did not have an application to make easier the job, instead they were using Excel spreadsheets which make the management process more complex and tedious since there is too much data for an engineer to fill.

Having this said the NE-C3 came with the great idea of creating a Web application to manage the content document data to create successful images. This is where they requested me to help the LCS project in their plan to manage all the systems that operate in launch services by creating a Web Application.

Description of activities

During the internship, I worked together with another intern from the Undergraduate Student Research Program, Jake Hochstadt with a similar accomplishment, to build a Rails application that can satisfy all the needs of the System Software group. Basically, the application is in a secured domain in which only users who are in the center can access the application successfully. It also has a sign in page which authenticates users from the NDC domain and does the same job prompting for the username and the password, as if was signing in into a NDC domain computer or NASA e-mail. We do not want all the NDC users to log in into our application, we do not even want all the NDC users from the Kennedy Space Center to access, so we filtered for only certain users can access the application. In our meetings with the clients we established that users will have certain roles to manage the application. The main users are known as "root", which have the ability to manage all in the application, including the addition or deletion of users in the application. The other group of users is the "admin", which can run around the whole application but cannot create, edit or delete any data in the server. The last group is called the "users" which have a similar role as the "admins", but with the exception of not seeing classified information, such as IP Address, Serial Numbers, Content Document ID's, and so forth. In short words, the application is filtered for users that are not allowed to see privileged information.

Sign in page

The image shows a screenshot of a web application's sign-in page. The background is a solid dark blue. At the top, the text "Content documents" is displayed in a white, sans-serif font. Below it, "Sign In Page" is also in white, sans-serif font. There are two white input fields: the first is labeled "Username" in small white text above it, and the second is labeled "Password" in small white text above it. Below the password field is a small, rectangular button with the text "Sign in" in white. The entire form is centered on the page.

Sign In Page creates a session to use the application.

When entering the URL, if you are not signed in, it will take you to the sign in page. This login session will only recognize NDC users who have access to the application. When the sign in is successful, it will take you to the home page, which is also the index page that contains the list of all the sets.

Home Page

Name	Description	Edit	Destroy
LDS	LDS	Edit	Destroy
ADS	ADS	Edit	Destroy
FR-1	Fire Room 1	Edit	Destroy
FRD	FRD	Edit	Destroy

The home page displays a message for the users to know that they are using U.S. Government property.

The first page, as all the other pages contains a header, which lets you choose the class that you want to see the information. Some of the information contained in the header are the Sets or operational rooms, the Workstations or computers, the Content Documents, the Software's, the Users, who are in charge of the Workstations, an Advanced Search that shows all the content documents that have a specific software or a group of software's, and finally, the Administrators, who are the users of the application.

The first page also displays a notification message for the NDC users. This message explains that the computers used to run the application are from U.S. government property and the users would be monitored for security reasons. As soon you close the message, it shows the whole page.

Log In

LOGIN STATUS

Signed in as
Root

Not you?
[Sign out](#)

Login status shows the user who is authenticated and an option to sign out.

Right now it is signed in as the Root user with a root role, who was created as an example of how the authentication works. Eventually the application has real NDC users with different types of roles. The sign out link deletes the session with the user and takes you to the login page again.

Set Show Page

Sets Workstation Content Documents Software Advanced Search Users Admin

Set Name: ADS

Description: ADS

Workstations:

- [hostname1](#)
- [hostname2](#)
- [hostname4](#)
- [hostname7](#)
- [hostname8](#)
- [hostname10](#)

[Edit Set](#) | [Back](#)

Javier Hochstetler and Ricardo Muniz

The set show page contains the name of the set, a short description, and the workstations that belongs to the set..

of the set, a description and, if there is any, the workstation(s) that is/are assigned to that specific set.

Obviously, we had to assign the same set in each workstation that appears in the list below in order to show which ones are contained in that set.

The actions to be chosen in this page are to see the information of a set, create a set, edit a set or delete a set. Each class have the same options in their own index page, so you can also show, create, edit or delete Workstations, Content Documents, Software's and Administrators. When clicking the name of a set, it shows the information of that specific set. Set fields contain the name

Content Document Show Page

Content Documents instance are similar to the sets but with more fields. The content document name is represented with a format of ###-###-###-###-###. Besides having other fields, the content document also contains the list of the software's that eventually the workstation will have.

Sets Workstation Content Documents Software Advanced Search Users Admin

Name: CCC-OSI-CD-AAA-111

Image: CCC-OSS-AAA-111

Test Document: N/A

Generation Document: N/A

Security Report: N/A

Attachment: N/A

Software Installed:

- [Adobe Flash Player Version 10](#)
- [Adobe Flash Player Version 9](#)
- [Adobe Reader 8.1.3](#)
- [Adobe Reader 9](#)
- [Excel](#)

1 2 Next >

Search Software...

[Edit](#) | [Back](#) | [Log](#) | [Software List](#)

[Download Excel Spreadsheet](#)

Menu Navigation

- SEARCH CONTENT DOCUMENTS
- NEW CONTENT DOCUMENT
[New Content Document](#)
- LOGIN STATUS
Signed in as **Root**
Not you?
[Sign out](#)

The Content Documents show page contains the name, an image associated to the Content Document, additional information such as test document, generation document and security report. Finally it contains all the software attached to that specific Content Document in paginated form.

Search Software

Software Installed:

- [Adobe Flash Player Version 10](#)
- [Adobe Flash Player Version 9](#)

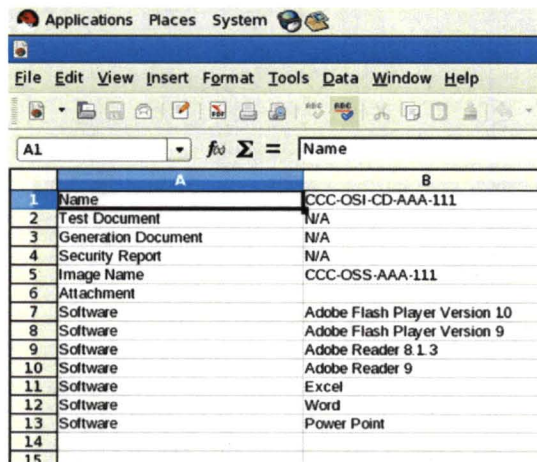
Adobe F

Software search in a single Content Document.

If the software list is huge, you can do a search for software in that content document. As soon as you start typing, it begins to search for software's with what is typed in the box. This helps the user to search for software that maybe doesn't belong to the content document and you want to delete from it or for software that you want to add to the content document if the software does not appear in the search.

Excel Spreadsheet

We also have an option to download the values of a Content Document into an Excel Spreadsheet. So to do that, we just click on the link called "Excel Spreadsheet" and that makes all the work.

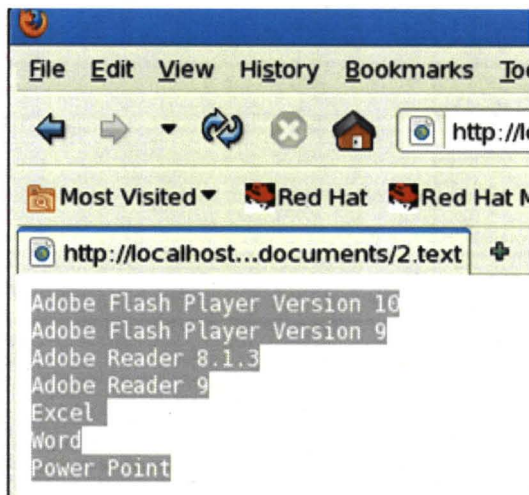


The screenshot shows an Excel spreadsheet with the following data:

	A	B
1	Name	CCC-OSI-CD-AAA-111
2	Test Document	N/A
3	Generation Document	N/A
4	Security Report	N/A
5	Image Name	CCC-OSS-AAA-111
6	Attachment	
7	Software	Adobe Flash Player Version 10
8	Software	Adobe Flash Player Version 9
9	Software	Adobe Reader 8.1.3
10	Software	Adobe Reader 9
11	Software	Excel
12	Software	Word
13	Software	Power Point
14		
15		

Excel Spreadsheet of a Content Document information.

Software List



Software list of a Content Document for easy copy and paste.

The software list link makes an html page with the list of the software's if wanted to copy in a text format. It provides the user a quick tool to copy and paste a list of software's from a content document.

Software Log

Next link is the log link, which is very useful because it makes a log for all the software's added and deleted from a content document.

CCC-OSI-CD-AAA-111

Action	Software	Created At	
Added	Accurev 4.7.2	2011-05-04 12:23:12 UTC	Show
Added	Active Perl 5.10	2011-05-04 12:23:12 UTC	Show
Added	Adobe Flash Player Version 10	2011-05-04 12:23:12 UTC	Show
Added	Adobe Flash Player Version 9	2011-05-04 12:23:12 UTC	Show
Added	Adobe Reader 8.1.3	2011-05-04 12:23:12 UTC	Show
Added	Adobe Reader 9	2011-05-10 14:08:52 UTC	Show
Added	Excel	2011-05-10 14:08:52 UTC	Show
Added	Power Point	2011-05-10 14:08:52 UTC	Show
Added	Word	2011-05-10 14:08:52 UTC	Show
Removed	Accurev 4.7.2	2011-05-10 14:08:52 UTC	Show undo
Removed	Active Perl 5.10	2011-05-10 14:08:52 UTC	Show undo
Destroy			

[Back](#)

Action List of a Content Document to keep track of added and removed software.

You can also clear a log if desired with the destroy link, but the most efficient tool it have is to undo a deleted software. To explain this better, let's go into a situation. Let's say we are editing a content document and we want to remove some of the software's it has. So we go to the edit action and remove a couple of software's. So assuming that we cleared the log before deleting software's, this is how Action List should look like.

CCC-OSI-CD-AAA-111

Action	Software	Created At		
Removed	Excel	2011-05-10 14:29:37 UTC	Show	undo
Removed	Power Point	2011-05-10 14:29:37 UTC	Show	undo
Removed	Word	2011-05-10 14:29:37 UTC	Show	undo
Destroy				

[Back](#)

A demonstration of the undo feature that brings Action List to a Content Document.

We can see that the log says that we removed three Adobe software versions, but let's say we want Adobe Flash Player Version 10 added back to the Content Document software list. To the right of the description log, notice that for the removed software we have an 'undo' action. As soon as you hit the 'undo' link, it will put back the software into that content document again and also creates a new log saying that we added back the Adobe Flash Player Version 10.

CCC-OSI-CD-AAA-111

Action	Software	Created At		
Removed	Excel	2011-05-10 14:29:37 UTC	Show	
Removed	Power Point	2011-05-10 14:29:37 UTC	Show	undo
Removed	Word	2011-05-10 14:29:37 UTC	Show	undo
Added	Excel	2011-05-10 14:33:28 UTC	Show	
Destroy				

[Back](#)

Added back the software to the Content Document with the undo link.

Software Page

The Software instance has the list off all the software's that are essential in the workstations. Since the list of software's is huge, we implemented a search for software's. As soon as you click the search button it searches for software's that are similar to what the user inputs.

Name		
Adobe Flash Player Version 10	Edit	Destroy
Adobe Flash Player Version 9	Edit	Destroy
Adobe Reader 8.1.3	Edit	Destroy
Adobe Reader 9	Edit	Destroy
Adobe Reader 9.1.0	Edit	Destroy

As you can see, we used the search feature for software in which we wanted to search for all the software with the name of 'Adobe'. The search displays the list of software with 'Adobe' on the name.

Advanced Search

[Add software](#)

[Search](#)

[New Search](#)

To access the Advanced Search, click on the Advanced Search link displayed in the header of the application.

The Advanced Search is a feature that allows searches for all the Content Documents that contain a selected list of software. To access the 'Advanced Search' feature, you will need to click on the link provided in the header section. The search starts with no software added. As soon as you press the 'Add Software' link, it provides a selection box with the list of all the software listed in the application. You can add as much software as you want, but be aware that for a Content Document to show it needs to have all the software's you selected.

Advanced Search

[Add software](#)

Adobe Flash Player Version 10	<input type="checkbox"/>	remove
Adobe Flash Player Version 9	<input type="checkbox"/>	remove
Adobe Reader 8.1.3	<input type="checkbox"/>	remove
Adobe Reader 9	<input type="checkbox"/>	remove

[CCC-OSI-CD-CCC-333](#)

[CCC-OSI-CD-BBB-222](#)

[CCC-OSI-CD-AAA-111](#)

[New Search](#)

The four software's selected are contained in the three Content Documents displayed.

When the software or a list of software is selected, just click the 'search' button and it will display all the Content Documents with the software selected. Also the Content Documents are linkable, so if you click a Content Document you can go directly to add software you want or delete software that you do not want from the Content document. This feature was created because it helps the user to search for a particular group of Content Documents since the list of Content Documents and software's is huge and we want the application to be as much user friendly as it can be.

Roles for Administrators

Last but not least, we have the Administrators of the application. As explained before, Administrators have different roles, ones can manage everything and others do not have access to some features. Let's say we create a user with the User role privileges. This user does not have access to create, edit destroy information and also cannot see IP addresses or Serial Numbers. This is done because we do not want all the users to have access to the server and create, edit or delete information without permission. Also it filters the classified information to be seen by only users to are able to see it.

New admin

Name

User

Username

username

Role

User

Password

.....

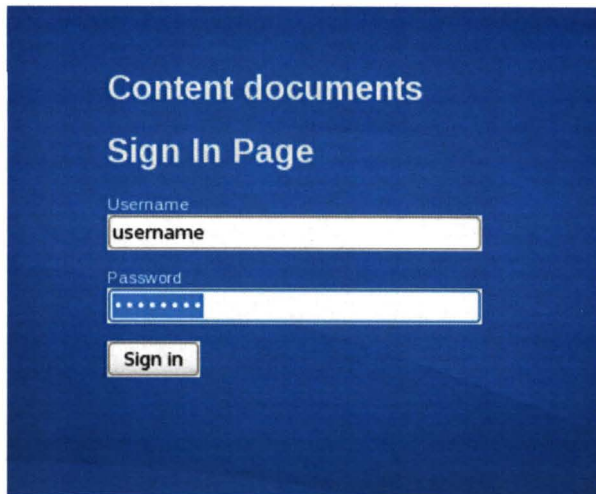
Password confirmation

.....

Create Admin

[Back](#)

Creating a new administrator with a user role.

The image shows a web application interface with a dark blue background. At the top, the text 'Content documents' is displayed in a light blue font. Below it, 'Sign In Page' is written in a white font. There are two input fields: the first is labeled 'Username' and contains the text 'username'; the second is labeled 'Password' and contains seven dots. Below these fields is a button labeled 'Sign in'.

Signing in as the new user created.

So we finish creating the user and then we sign out as the current user. This action will destroy the current session and take you back to the sign in page. Then we enter the credentials of the new user we created. Because of the role we added to this new admin he does not have much privilege in the application. When you try to do an action that is not permitted, it displays a message telling the user that is not authorized to access the page or action and it redirects to the home page. This feature increases security in the application.

Results

We accomplished to have an application which can manage content documents information in a server. We added login functionalities for the NDC users that are allowed to use the application. We made important features that makes the work easier to the user, for example Advanced Search, Action List Logs, adding different roles and privileges for each user and so forth. I think that we did a good job in a fifteen week period internship.

Conclusion

I think that doing an internship in the Kennedy Space Center gives the experience of hard work in a real environment, and increases reputation of an individual in a good way to have success in finding a good job or even increasing the chance of working permanent at NASA in a future. It also helps to increase self esteem and to believe in you always. Finally, it makes proud your family, your friends and your school and motivates other students to give their best in their studies no matter what field they are.